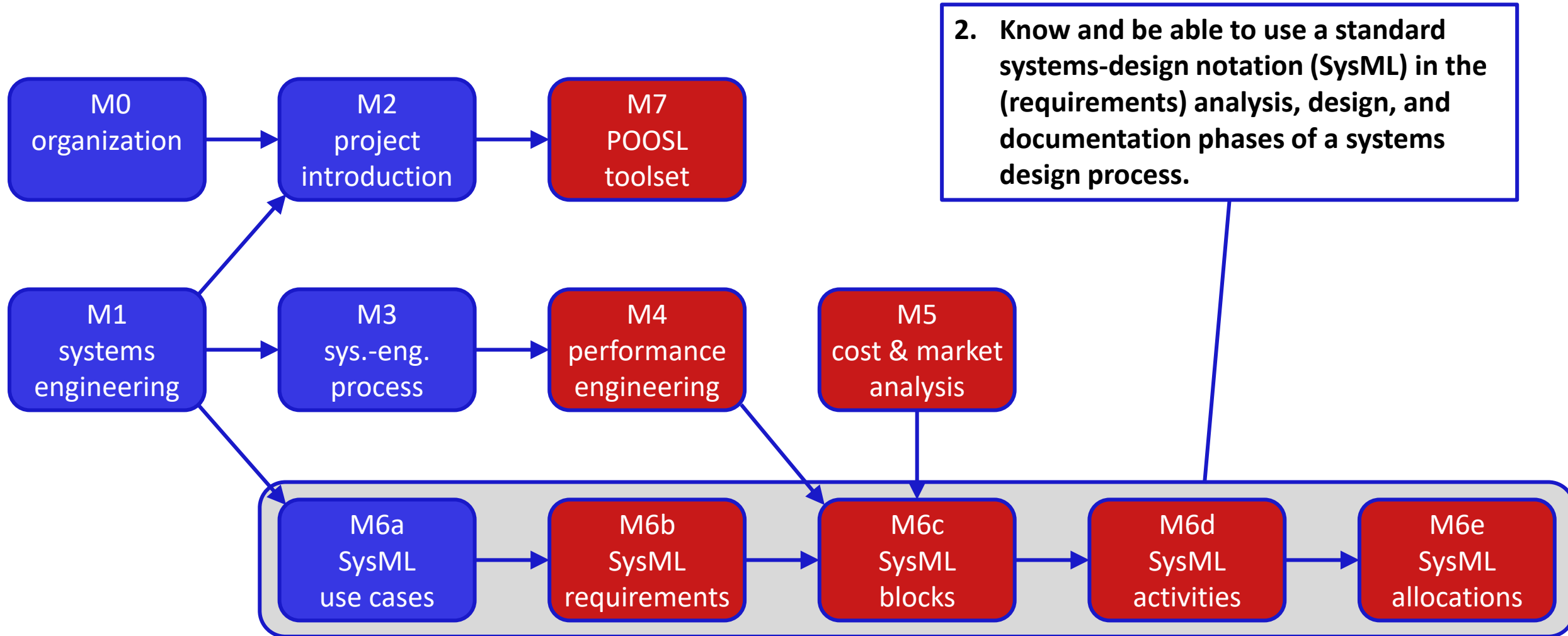


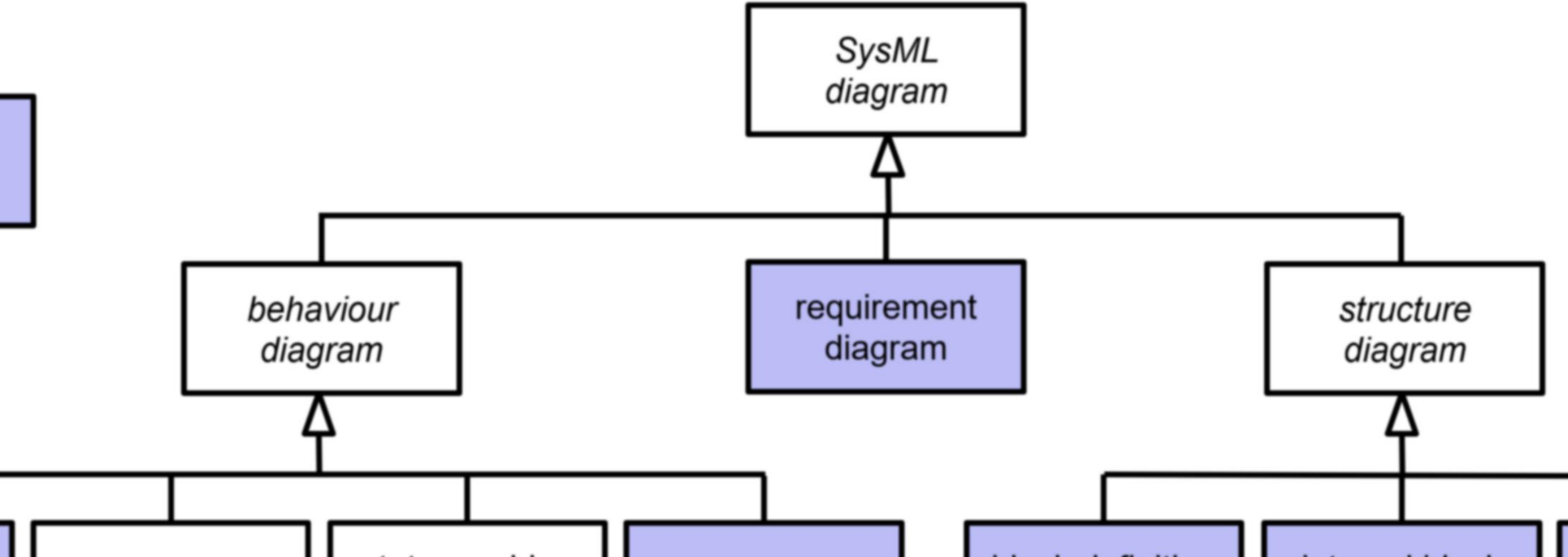
# 5XIC0 Electronic-Systems Engineering

Twan Basten, Martijn Hendriks

Electrical Engineering

# modules





## M6b – SysML requirements

5XIC0 Electronic-Systems Engineering

Martijn Hendriks

Slides in part based on a slide set of Kees Goossens and Dip Goswami

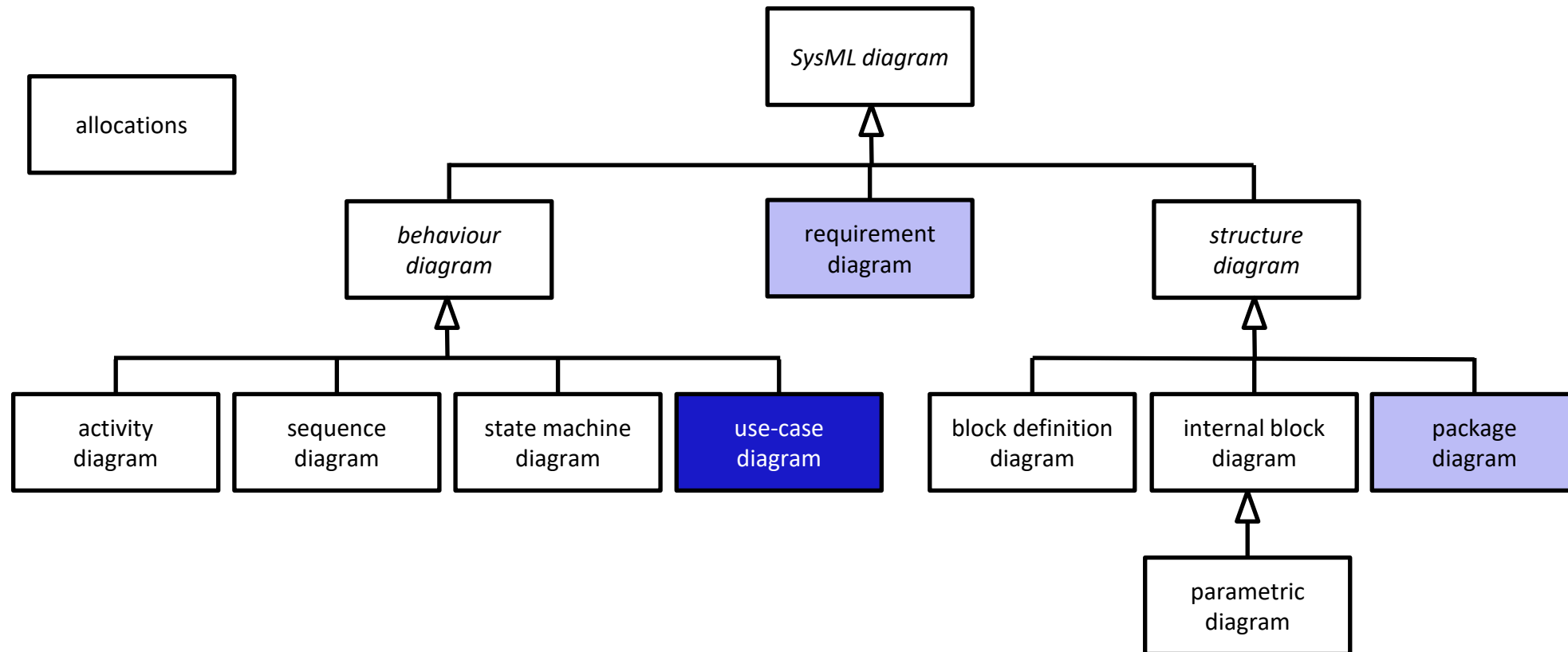
parametric  
diagram

# in this lecture

- SysML packages
- SysML requirements

# SysML – diagram overview

diagrams are **views** on the model  
(i.e., on a subset of **model elements**)



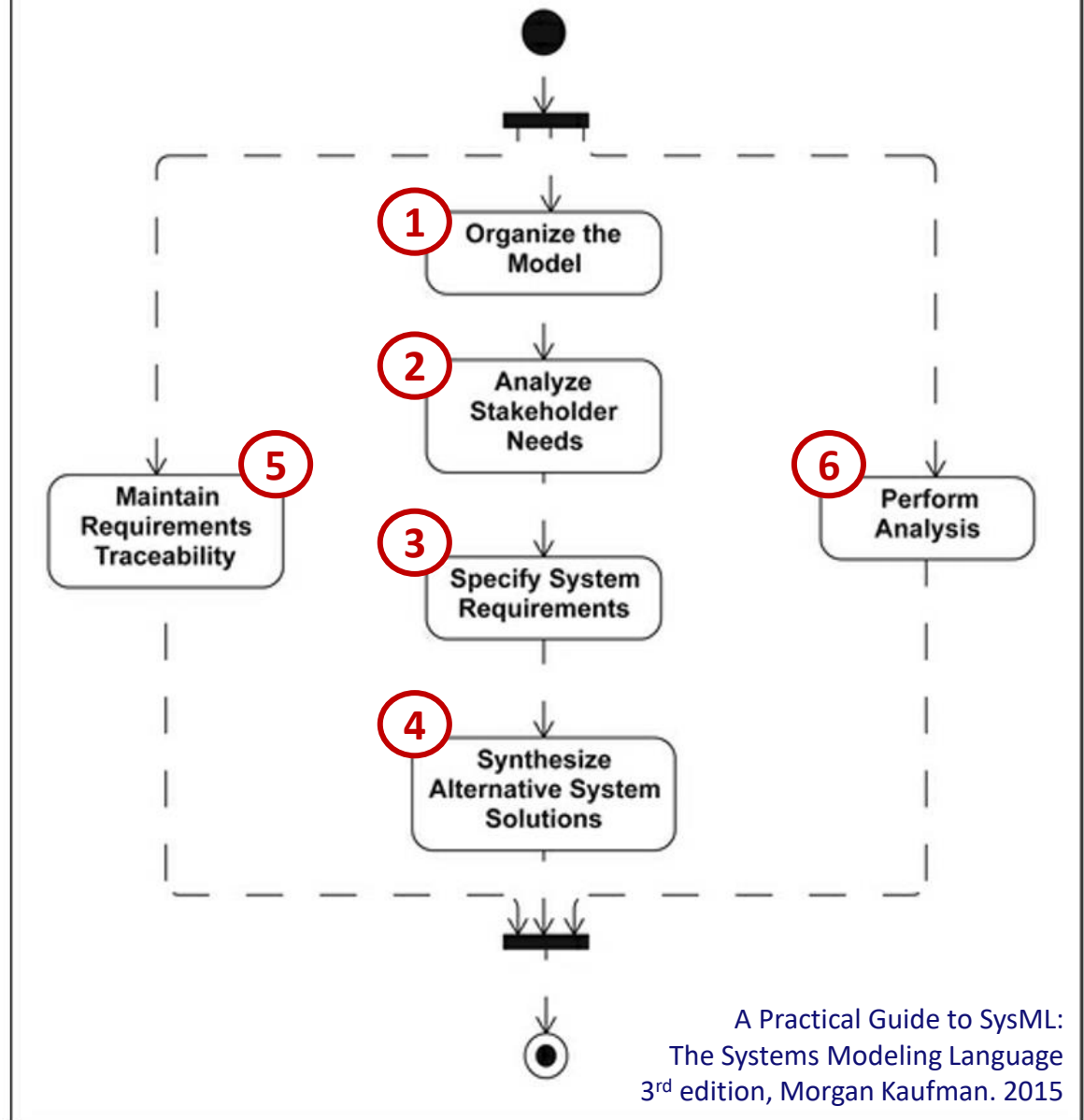
# SysML – packages – what & when

a model can contain **millions** of **model elements**

- each model element has a **container/parent**
- **packages** are **named** containers and are used to **organize** model elements in a **hierarchy**

# a simplified<sup>2</sup> MBSE method

1. SysML package diagram
2. stakeholders  
SysML UC diagrams, UC descriptions  
measures of effectiveness (moes)
3. SysML requirement diagrams
4. create multiple alternatives
  - SysML BDDs – system decomposition
  - SysML IBDs – interconnections
  - SysML Activity diagrams – UC refinements
  - SysML Allocations – activities to blocks
5. requirements tracking
  - SysML Allocation – reqs to blocks/activities
6.
  - SysML PAR diagrams – covering all moes
  - POOSL models – makespan
  - analytical model – profit
  - verification



# SysML – packages – what & when

a model can contain **millions** of **model elements**

- each model element has a **container/parent**
- **packages** are **named** containers and are used to **organize** model elements in a **hierarchy**

model organization happens **through the whole model lifecycle**

- having a well-organized model is very important

possible **organization principles**

- system hierarchy (system, component, sub-component, ...)
- process lifecycle (requirements, architecture, design, ..., operation)
- teams/disciplines working on the system (software, mechatronics, physics, ...)
- model-element kind (behavior, structure, requirements)
- ...



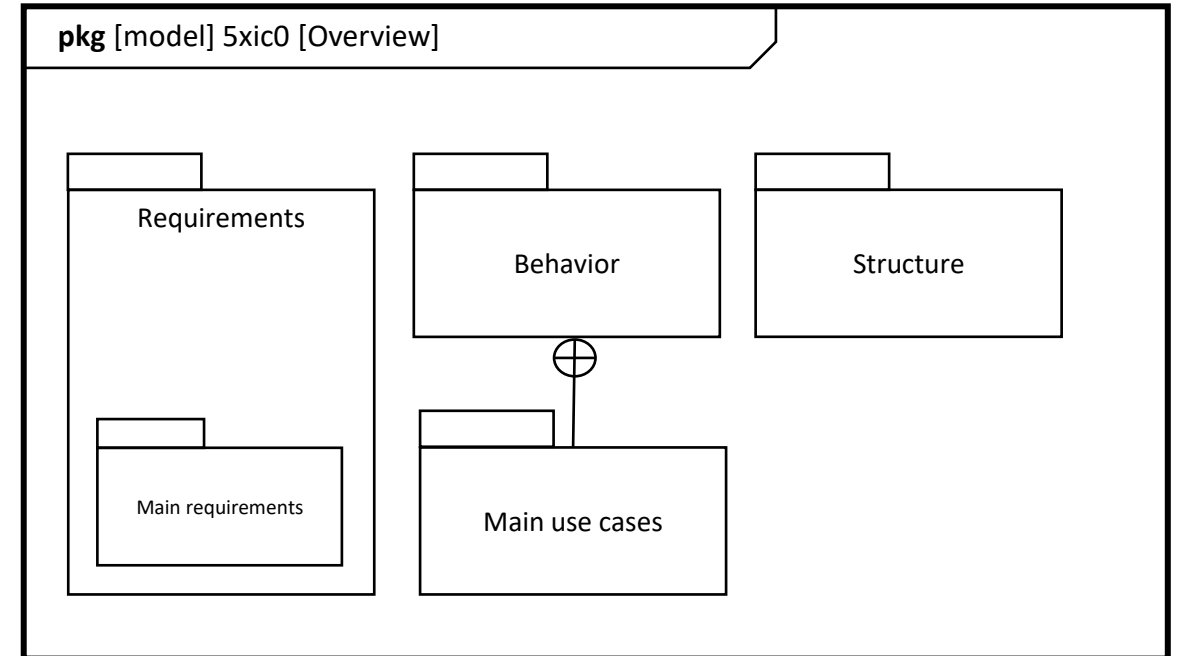
# SysML – packages – model elements & diagram (pkg)

model elements

- **package**: a named container for model elements

relations

- **containment**
  - by **nesting**
  - by **containment link**



suggested reading: section 6.1 – 6.4

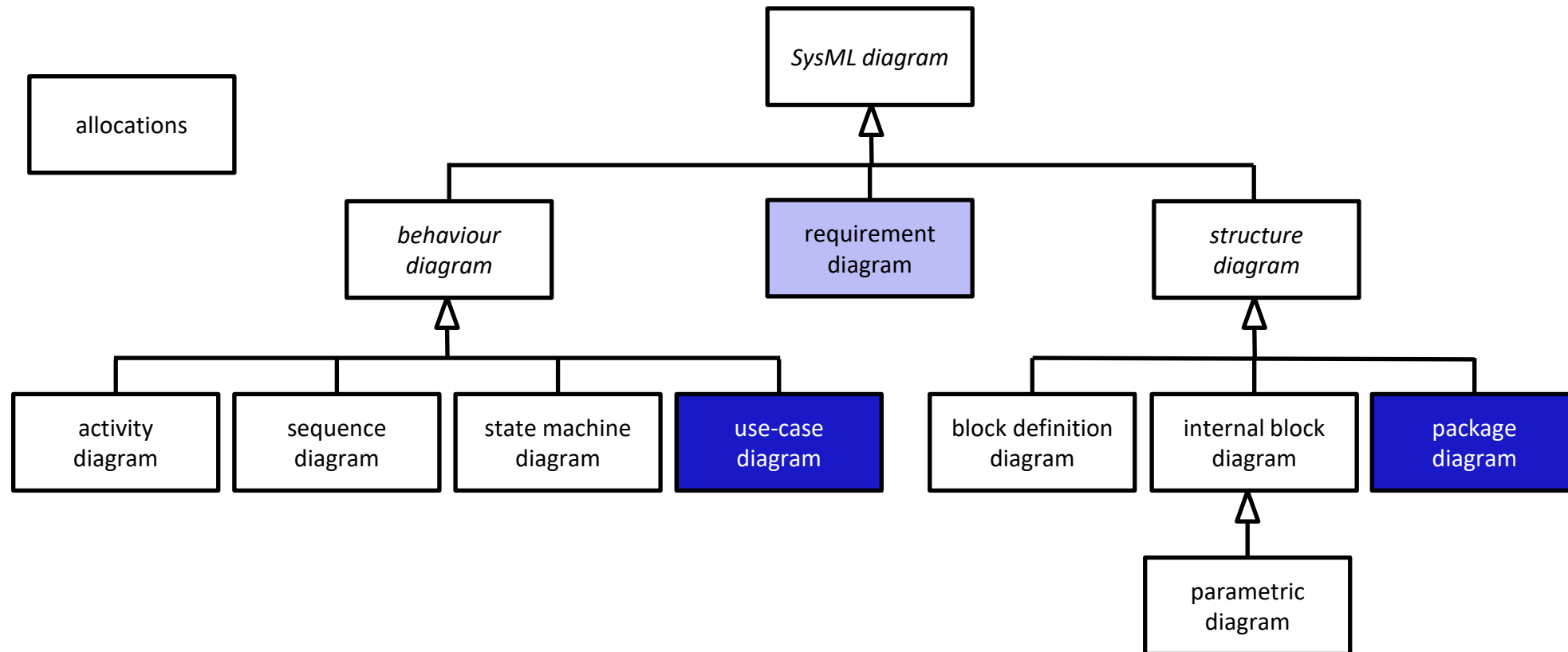
# think – pair – share

in your project group

- which top-level organization principle would you apply
- how can you work together on the model given that Papyrus does not support concurrent work (i.e., merging models is not possible)

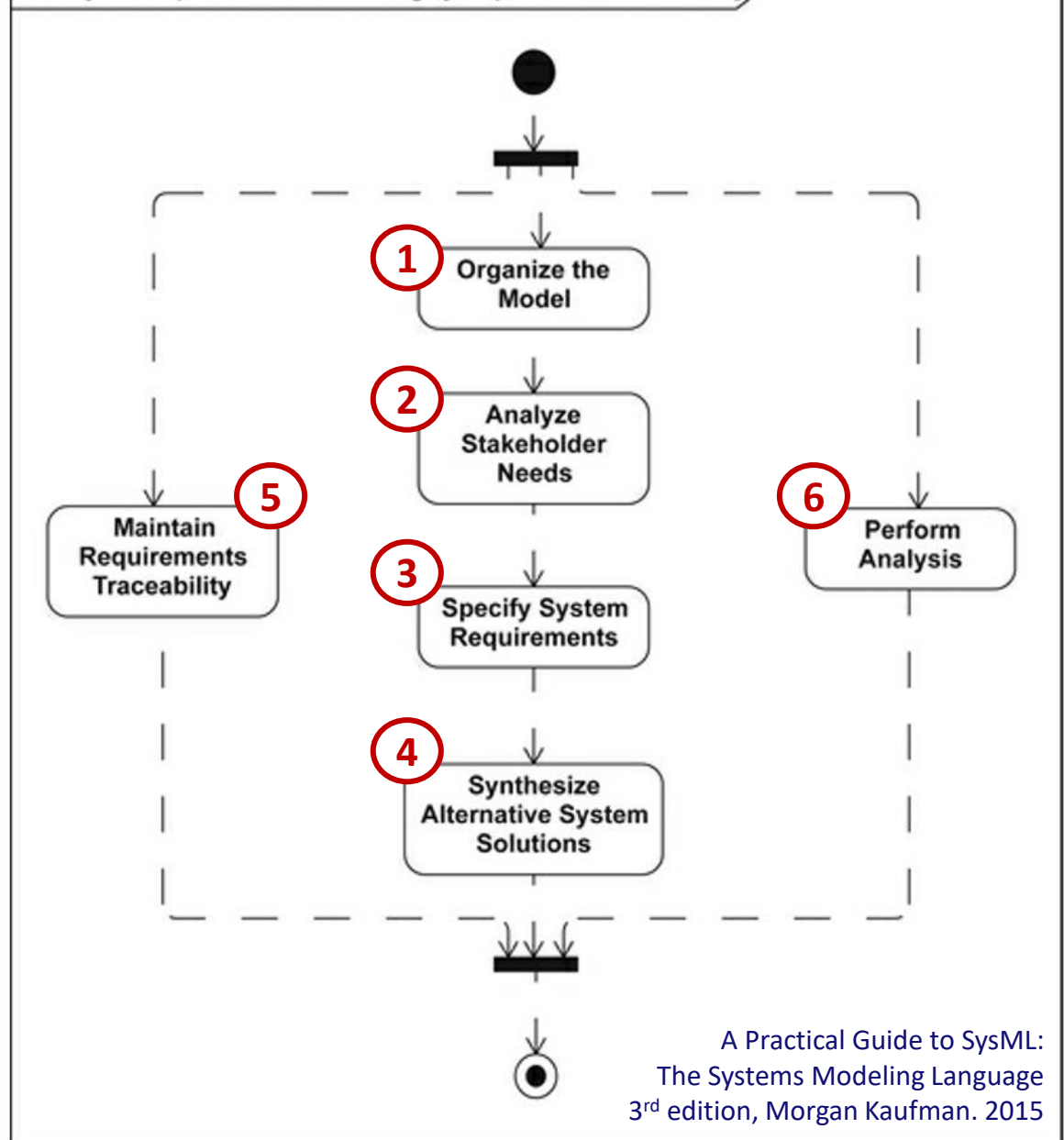
# SysML – diagram overview

diagrams are **views** on the model  
(i.e., on a subset of **model elements**)



# a simplified<sup>2</sup> MBSE method

1. SysML package diagram
2. stakeholders  
SysML UC diagrams, UC descriptions  
measures of effectiveness (moes)
3. **SysML requirement diagrams**
4. create multiple alternatives
  - SysML BDDs – system decomposition
  - SysML IBDs – interconnections
  - SysML Activity diagrams – UC refinements
  - SysML Allocations – activities to blocks
5. requirements tracking
  - SysML Allocation – reqs to blocks/activities
6.
  - SysML PAR diagrams – covering all moes
  - POOSL models – makespan
  - analytical model – profit
  - verification

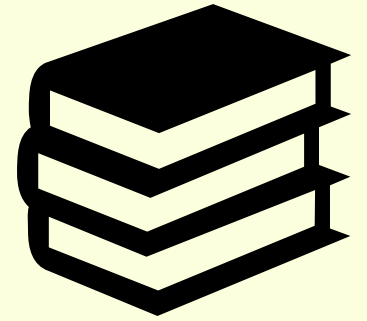


# requirements

a condition or capability needed by a user to solve a problem or achieve an objective

suggested readings (4 pages each)

- IEEE Guide for Developing System Requirements Specifications, IEEE Std 1233, 1998 Edition, Section 6. Well-formed requirements
- NASA Systems Engineering Handbook  
Appendix C: How to Write a Good Requirement - Checklist



# benefits of well-written requirements

good systems requirements **benefits** all subsequent phases of the **life cycle**:

1. **assurance** to the **customer**
2. early **bidirectional feedback** between **customer** and **engineers**
3. method to **identify problems** and **prevent misunderstandings**
4. **basis** for **system validation**
5. **protection** for **engineers**
6. support for **planning** design
7. assessing the effects of (inevitable) requirement **changes**



# SysML – requirements – what & when

requirement: a condition or capability needed by a user to solve a problem or achieve an objective

- come from many sources, e.g., in the car example
  - customers (top speed at least 130 km/h)
  - developing organization (re-use an earlier developed engine type)
  - government (emission control and safety)
  - ...
- **systems-engineering challenge** to make requirements
  - consistent (not contradictory)
  - feasible (can be realized)
  - sufficiently complete and validated (they reflect all stakeholder needs)
  - verified (system design and realization satisfy them)

# SysML – requirements – model elements

## model elements

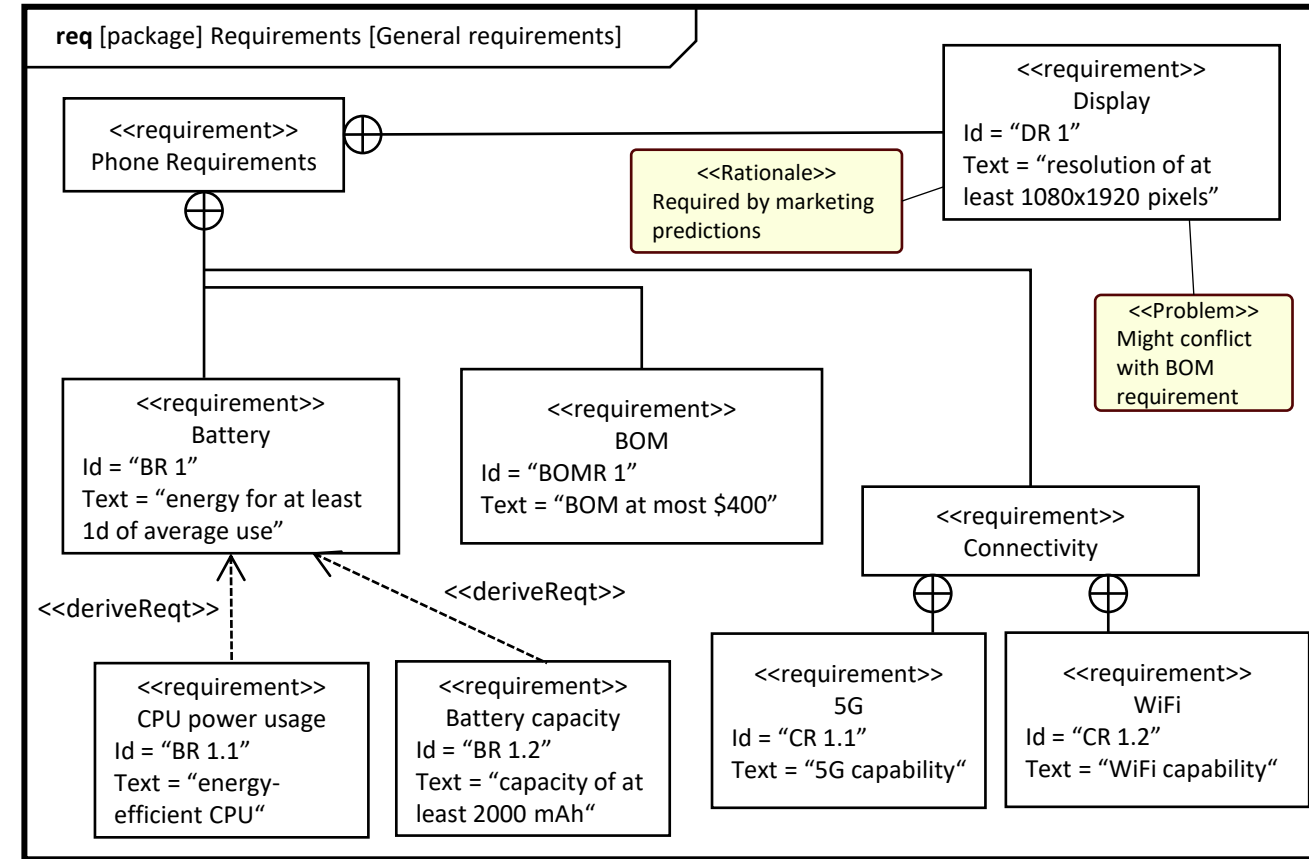
- **requirement**: name, identifier and textual description of the requirement
- **rationale**: reason for a particular decision
- **problem**: flag design issues

## requirement relationships

- **containment** for decomposition of requirements
- **deriveReq** for refinement of requirements, based on analysis

## relations to other model elements

- **satisfy, verify, refine, trace** are **cross-cutting** and explained in module on allocations

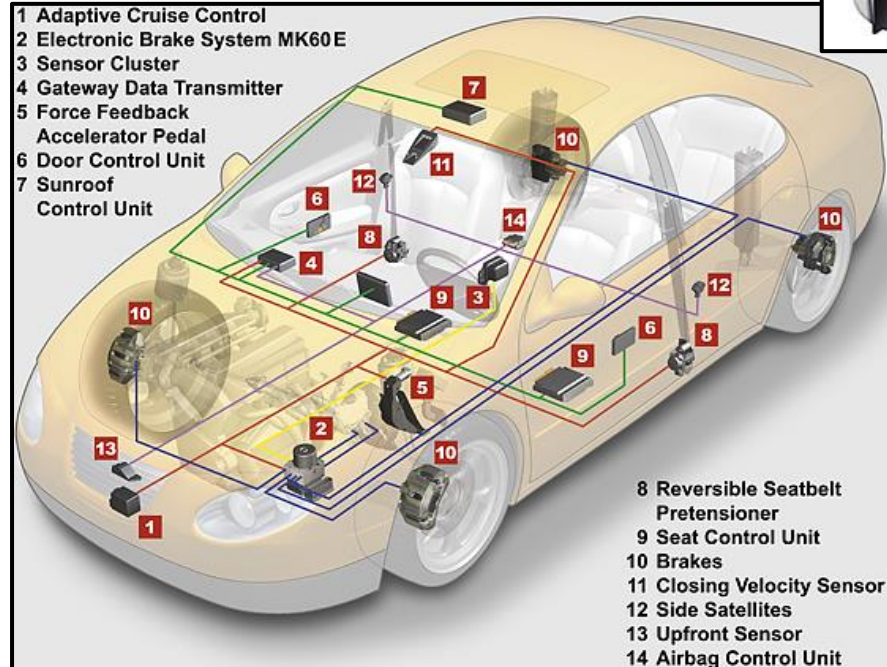
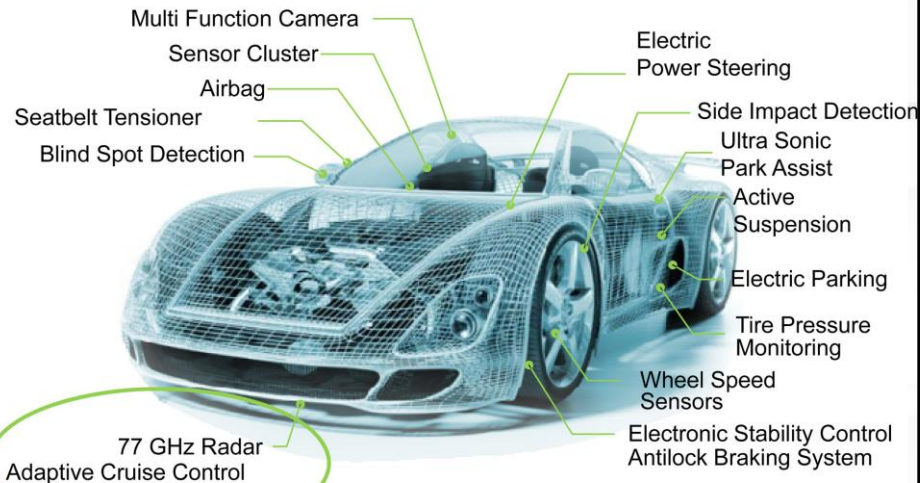
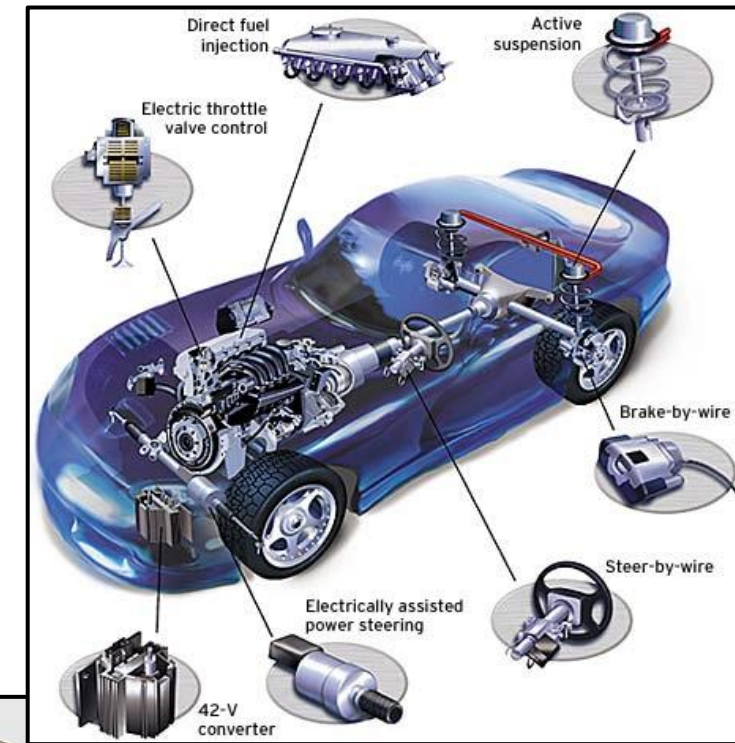


suggested reading:  
sections 13.1 – 13.3, 13.6 – 13.10



# SysML – running example

The requirements for our car can be decomposed into functional, performance, manufacturing and safety requirements. Top speed is at least 130 km/h, the doors should open wirelessly from within 30m. Furthermore, all passengers must be protected in a crash. This can be achieved by seatbelts (by law) and maybe by having airbags in each door. This clearly affects the BOM.



M6b - SysML requirements

sources: motorola, aa1car.com

# think – pair – share

- model the requirements

The requirements for our car can be decomposed into functional, performance, manufacturing and safety requirements. Top speed is at least 130 km/h, the doors should open wirelessly from within 30m. Furthermore, all occupants must be protected in a crash. This can be achieved by seatbelts (by law) and maybe by having airbags in each door. This clearly affects the BOM.

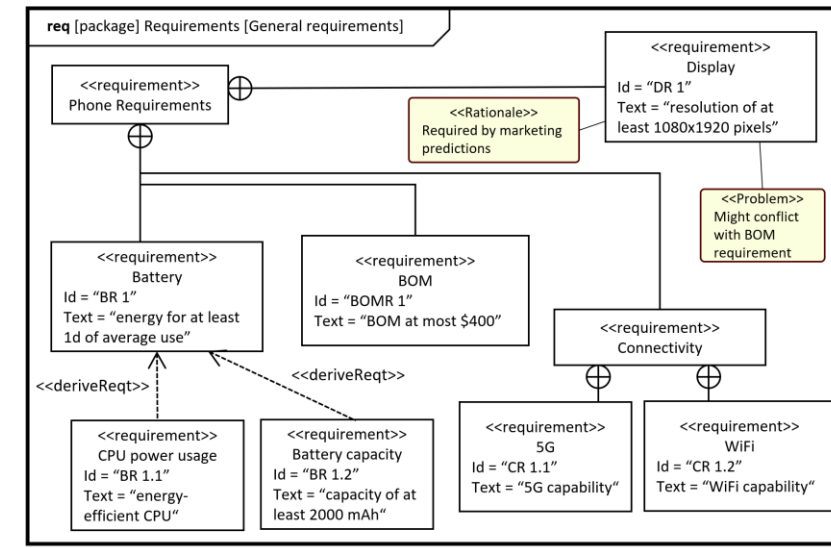
## SysML – requirements – model elements

model elements

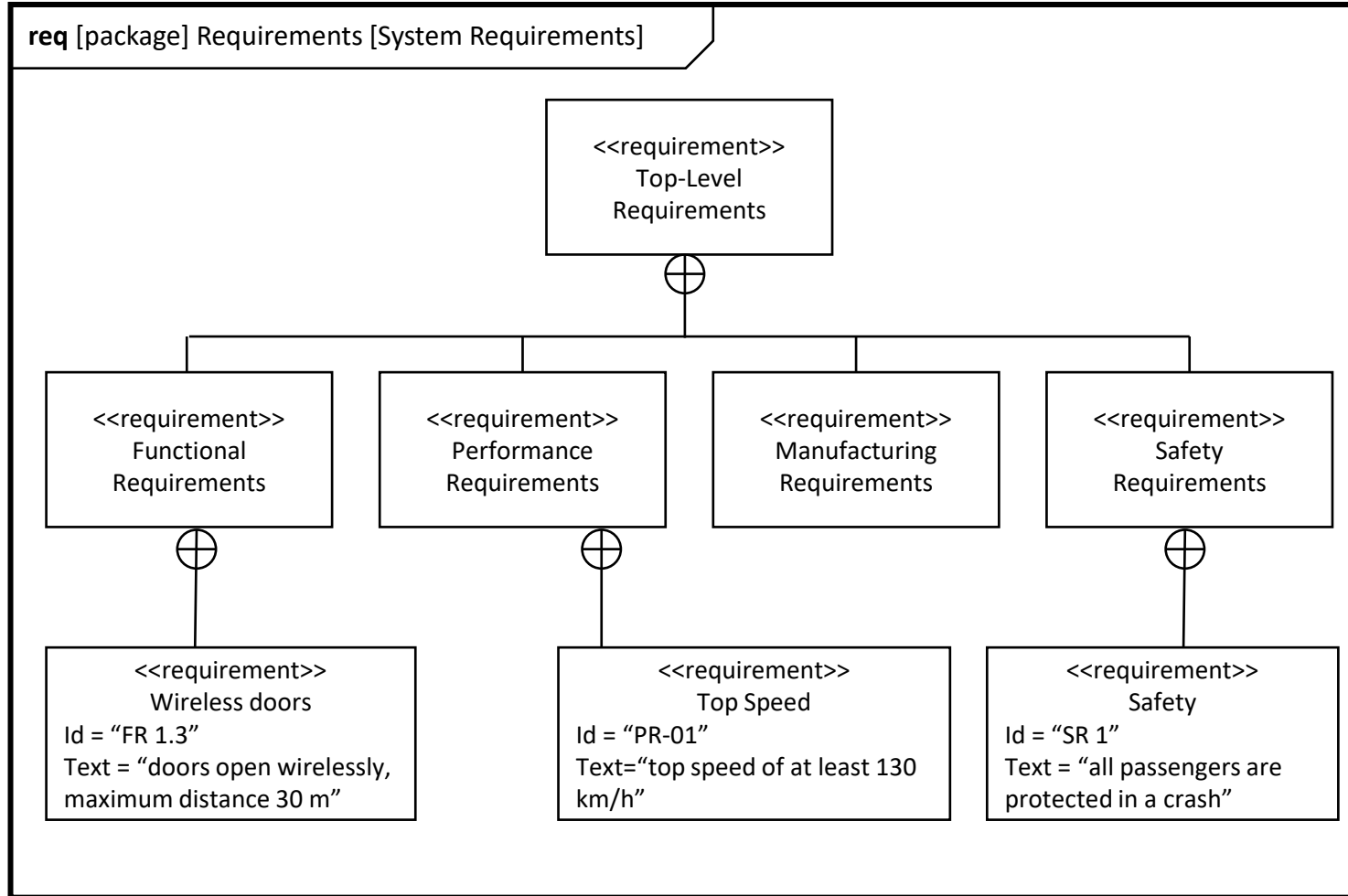
- **requirement**: name, identifier and textual description of the requirement
- **rationale**: reason for a particular decision
- **problem**: flag design issues

requirement relationships

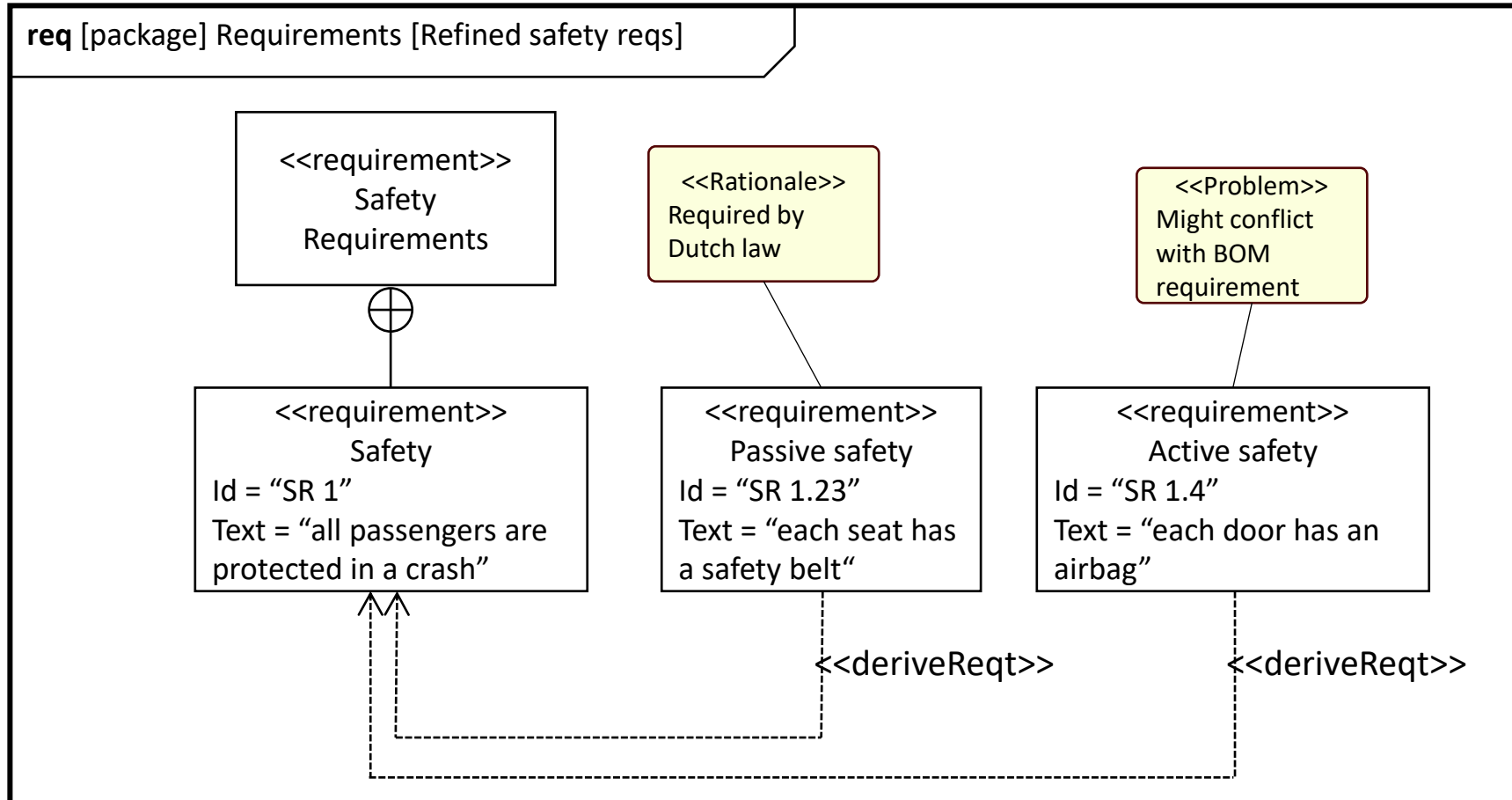
- **containment** for decomposition of requirements
- **deriveReq** for refinement of requirements, based on analysis



# SysML – requirement diagram (req)

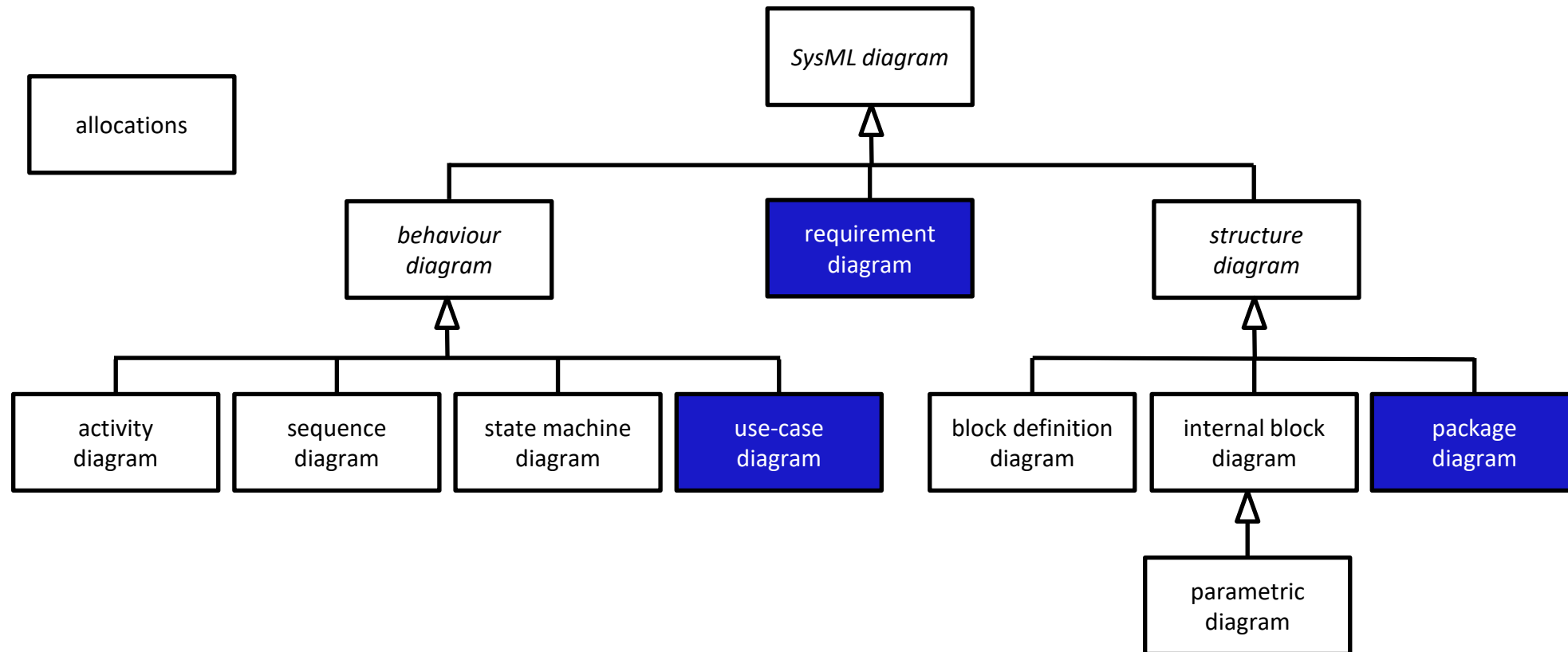


# SysML – requirement diagram (req)

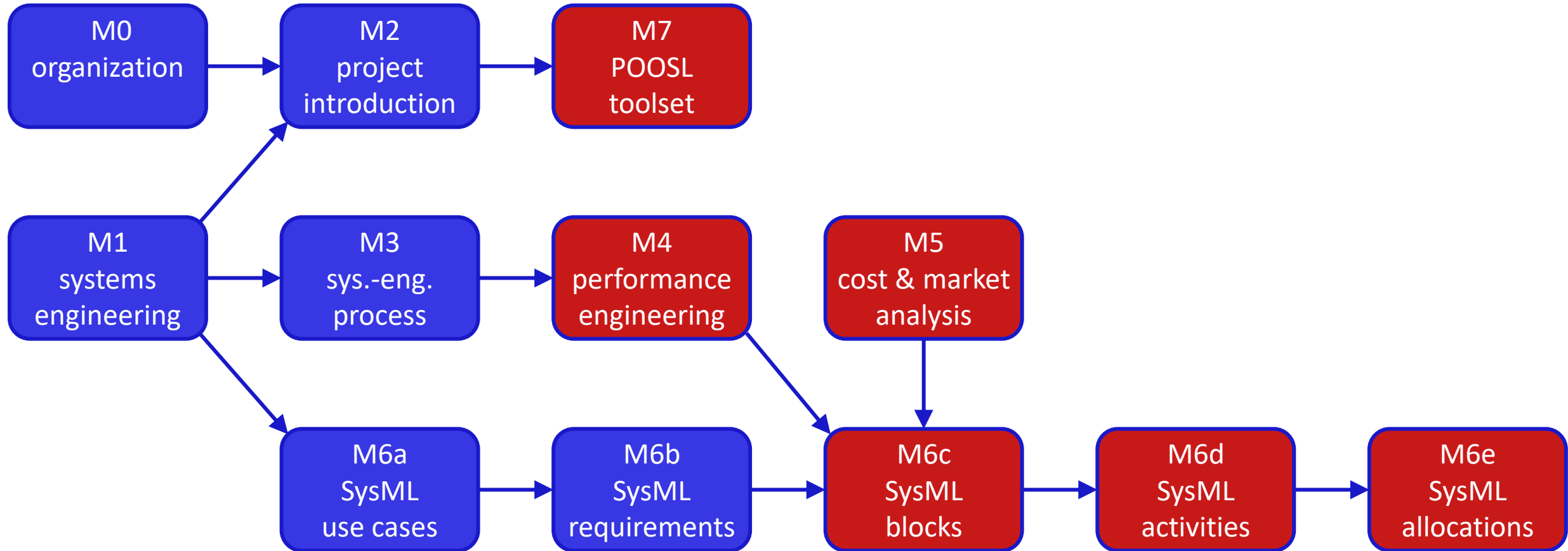


# SysML – diagram overview

diagrams are **views** on the model  
(i.e., on a subset of **model elements**)



# modules



## to remember

packages are for organization of the model elements in a hierarchy

several principles for model organization

requirements are the basis for the specification, validation and verification of a (sub)system

requirement derivation corresponds to a design decision

requirement tracking supports verification and is explained in M6e

# todo

- watch the Papyrus video on requirements